

Historie in StUF kennisgevingen

Uitgangspunten

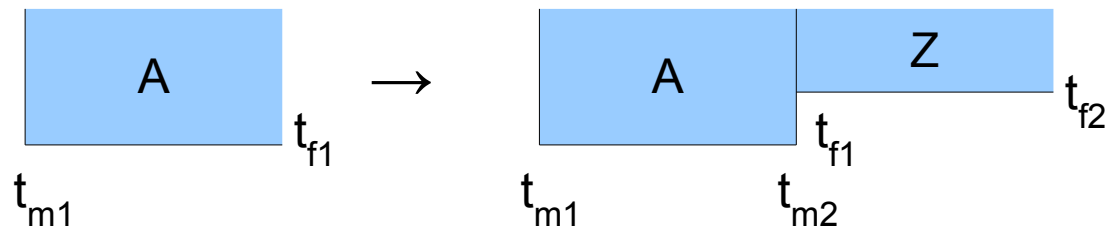
- Basiseenheid: een mutatie
- Ondersteuning materiële en formele historie
- Systemen kunnen onafhankelijk van elkaar muteren
- Kennisgeving: alleen mutatie actuele gegevens
- Correctie historische gegevens alleen in synchronisatieberichten

Wijziging actueel gegeven (1)

Eigenschap A die nu geldig is verandert op t_{m2} in Z en dit is op t_{f2} geregistreerd

Notatie: $A(t_{m1}, t_{m2}) \rightarrow Z((t_{m2}, \infty); t_{f2})$

Linked list representatie

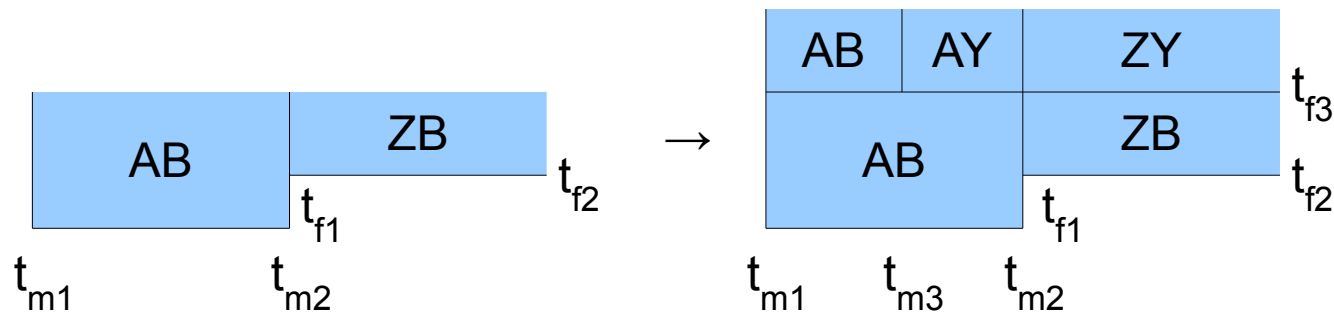


Discussie: Is t_{m1} relevant?

Wijziging actueel gegeven (2)

Eigenschap B die nu geldig is verandert op $t_{m3} < t_{m2}$ in Y en dit is op t_{f3} geregistreerd

$$B(t_{m1}, t_{m3}) \rightarrow Y((t_{m3}, \infty); t_{f3})$$



Discussie: Mag dit buiten synchronisatie?

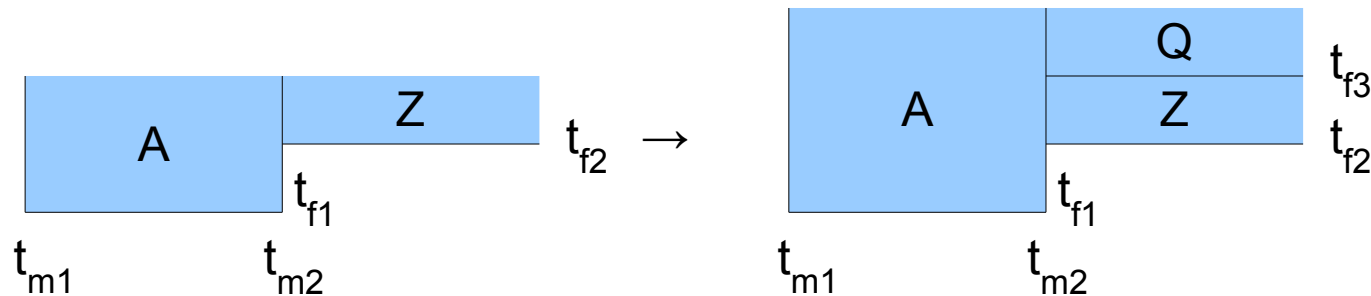
Mag dit buiten synchronisatie?

- **Identificatie**
Oud bevat actuele kerngegevens → Geen probleem
- **Onafhankelijkheid systemen**
Systeem dat B levert hoeft mutatie A niet te kennen
- **Implementatie sowieso nodig**
Verwerking synchronisatiebericht

Correctie actueel gegeven (1)

Eigenschap Z die nu geldig is wordt op t_{f3} gecorrigeerd in Q

$$Z(t_{m2}, \infty) \rightarrow Q((t_{m2}, \infty); t_{f3})$$

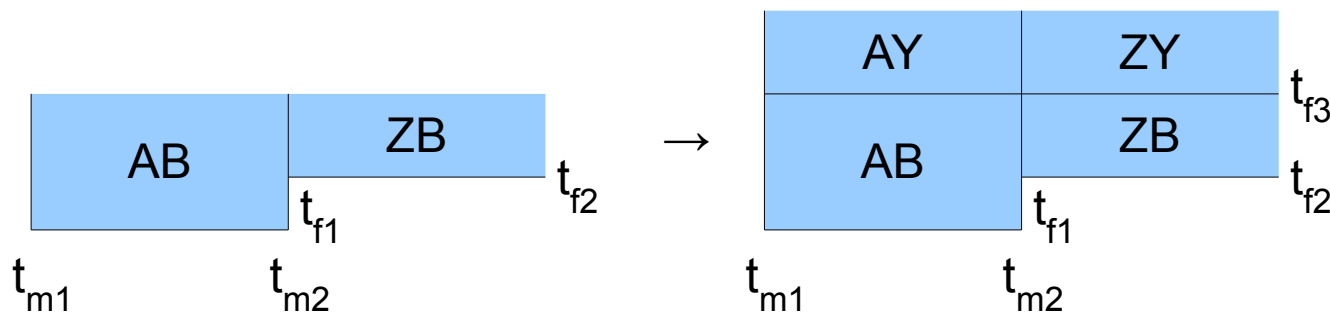


t_{m2} in oud is essentieel voor bepalen te corrigeren waarde
Z moet vóór t_{m2} een andere waarde hebben

Correctie actueel gegeven (2)

Eigenschap B die nu geldig is wordt op t_{f3} gecorrigeerd in Y

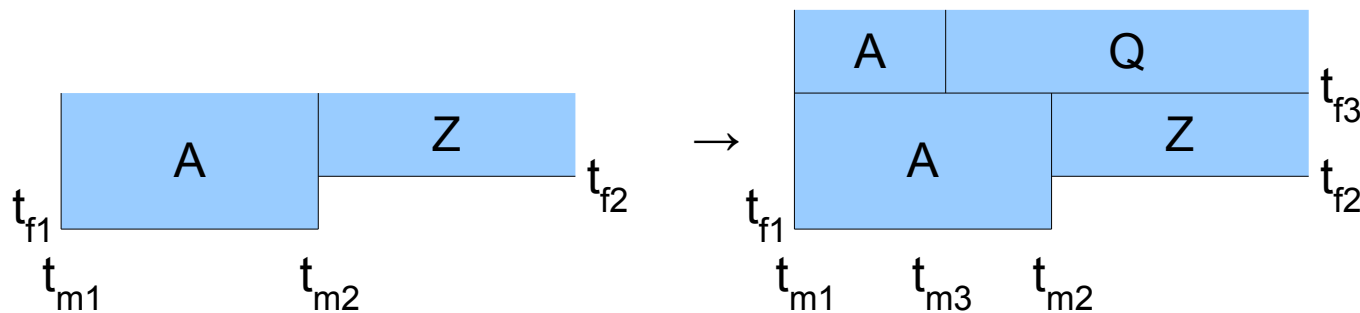
$$B(t_{m1}, \infty) \rightarrow Y((t_{m1}, \infty); t_{f3})$$



Correctie beginGeldigheid (1)

Z wordt op t_{f3} waarde gecorrigeerd in Q en
beginGeldigheid in $t_{m3} < t_{m2}$

$$Z(t_{m2}, \infty) \rightarrow Q((t_{m3}, \infty); t_{f3})$$

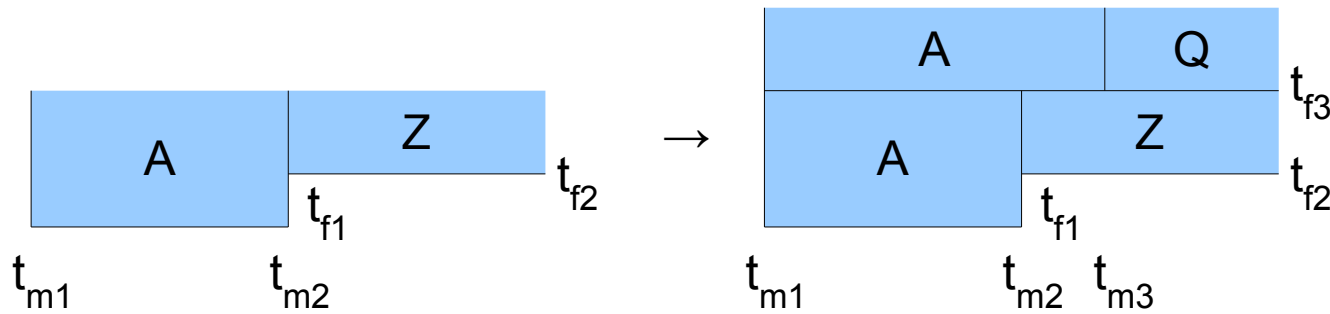


A wordt deels verdrongen door Q

Correctie beginGeldigheid (2)

Z wordt op t_{f3} waarde gecorrigeerd in Q en
beginGeldigheid in $t_{m3} > t_{m2}$

$$Z(t_{m2}, \infty) \rightarrow Q((t_{m3}, \infty); t_{f3})$$

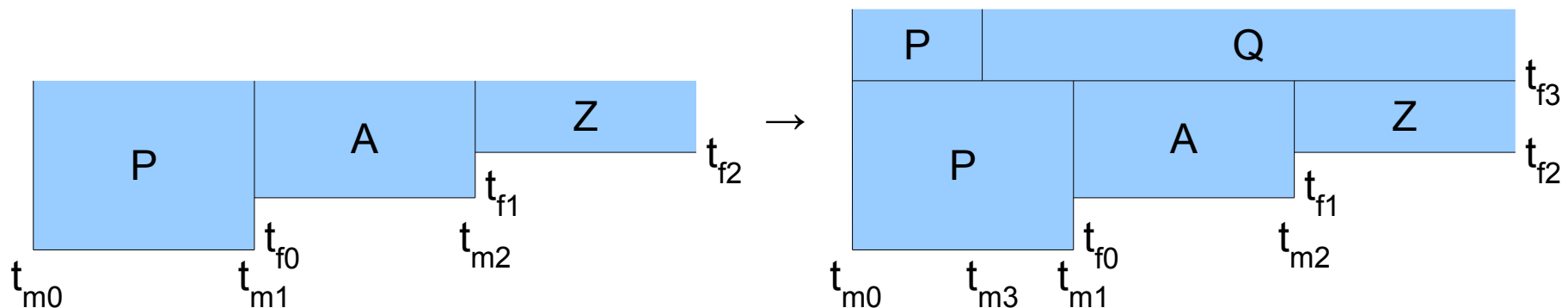


A volgt beginGeldigheid Q

Correctie beginGeldigheid (3)

Z wordt op t_{f3} waarde gecorrigeerd in Q en
beginGeldigheid in $t_{m3} < t_{m1}$

$$Z(t_{m2}, \infty) \rightarrow Q((t_{m3}, \infty); t_{f3})$$

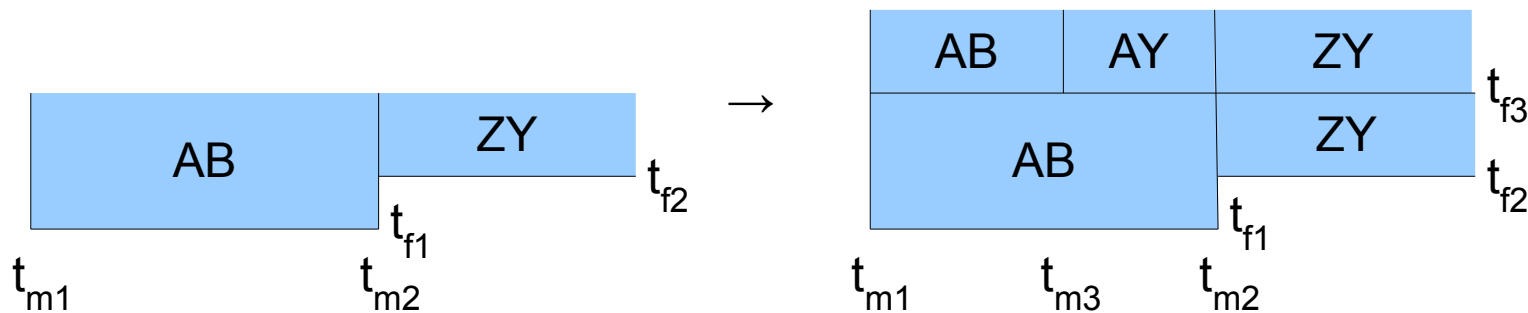


A wordt geheel en P deels verdrongen

Correctie beginGeldigheid (4)

Voor Y wordt op t_{f3} beginGeldigheid gecorrigeerd in $t_{m3} < t_{m2}$

$$Y(t_{m2}, \infty) \rightarrow Y((t_{m3}, \infty); t_{f3})$$

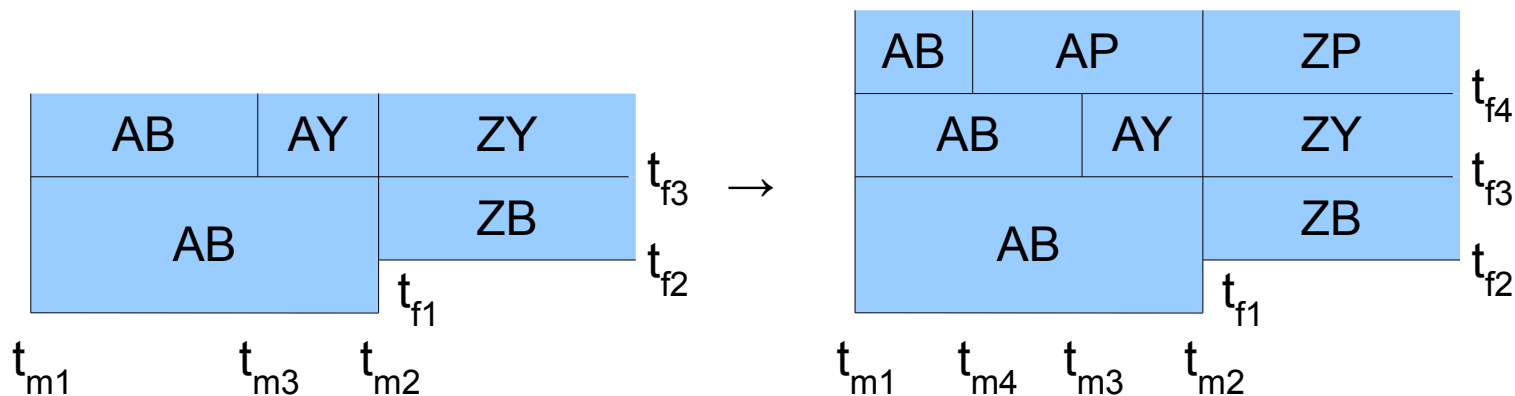


*Alleen voor gespecificeerde waarden die **wijzigen** op t_{m2} wijzigt beginGeldigheid (Denk aan de kerngegevens!)*

Correctie beginGeldigheid (5)

Voor Y wordt op t_{f4} waarde gecorrigeerd in P en
beginGeldigheid in $t_{m4} < t_{m3}$

$$Y(t_{m3}, \infty) \rightarrow P((t_{m4}, \infty); t_{f4})$$



Historische correcties

StUF-standaard

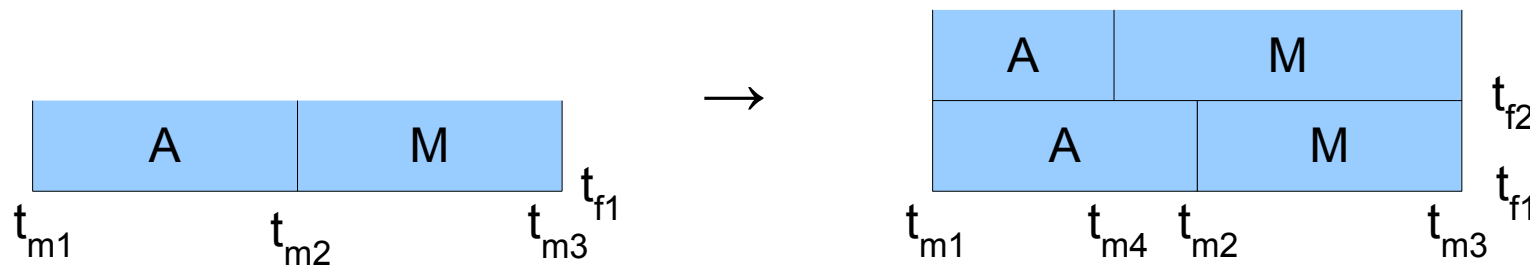
- In oud eindGeldigheid een geldige waarde
- Muv oudste beginGeldigheid alleen correcties van eindGeldigheid geoorloofd
- Bij correctie eindGeldigheid moet beginGeldigheid in oud/nieuw gelijk zijn
- Correctie eindGeldigheid historisch analoog aan correctie beginGeldigheid actueel
- Extra operatie: tussenvoegen historie

Corrigeren eindGeldigheid

$$A(t_{m1}, t_{m2}) \rightarrow A((t_{m1}, t_{m4}); t_{f2}),$$

A moet van waarde veranderen op t_{m1} en t_{m2}

$t_{m2} < \infty$ en t_{m4} mag ∞ zijn



∞ voor t_{m4} is nodig als bij een correctie een eindGeldigheid verschuift voor een element en niet verschuift voor een kerngegeven dat op eindGeldigheid ook wijzigt.

Binnen een correctie actueel kan dit niet gespecificeerd worden, omdat de kerngegevens daar verplicht zijn.

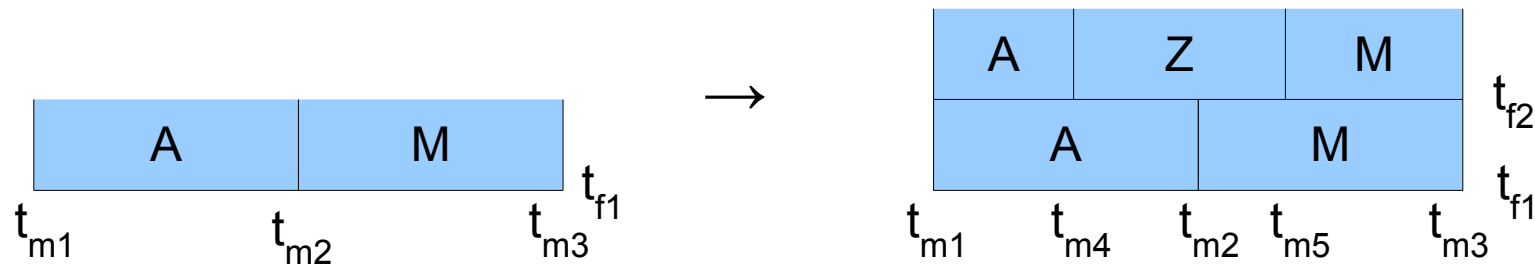
Binnen een synchronisatiebericht zijn de kerngegevens niet verplicht.

Tussenvoegen historie

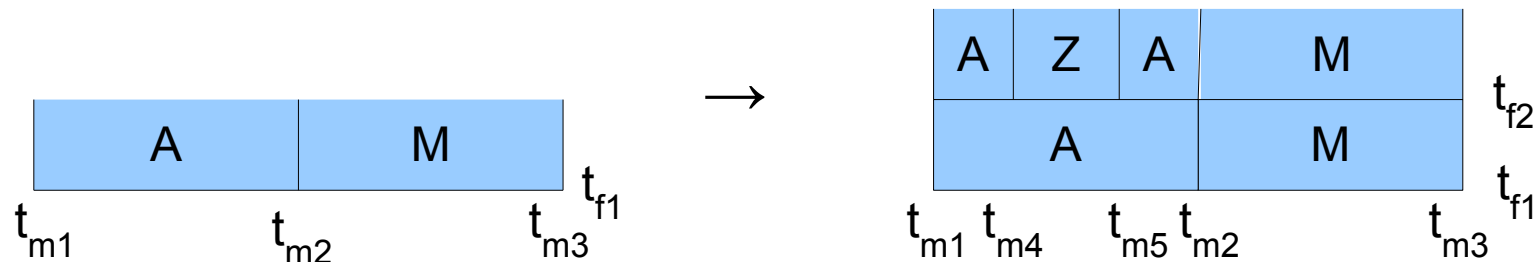
$$A(t_{m1}, t_{m2}) \rightarrow Z((t_{m4}, t_{m5}); t_{f2}),$$

$A \neq Z$ en $t_{m2} < \infty$ en $t_{m1} < t_{m4} < t_{m2}$ en $t_{m4} < t_{m5} < \infty$,

A moet van waarde veranderen op t_{m1} en t_{m2}



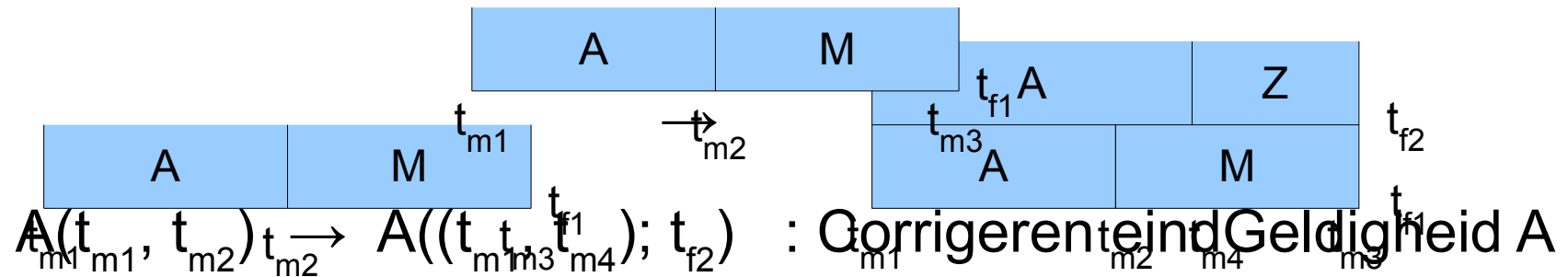
of



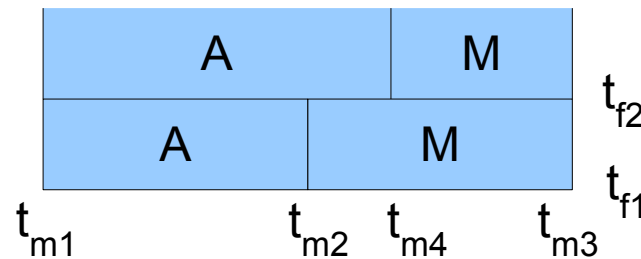
Opvullen met A voor t_{m1} tot t_{m4} en t_{m5} tot t_{m2} , als $t_{m5} < t_{m2}$

Verdringen door Z voor t_{m4} tot t_{m5} , als $t_{m5} > t_{m2}$

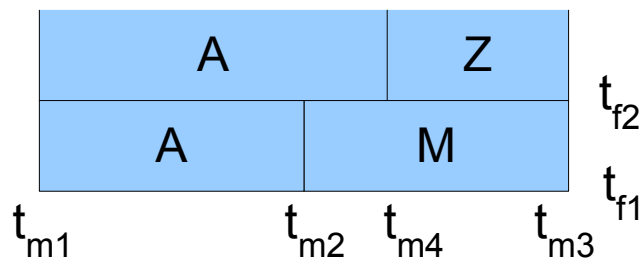
Vraag



???



$M(t_{m4}, t_{m3}) \rightarrow Z((t_{m4}, t_{m3}); t_{f2}) : \text{Materieel corrigeren } M \rightarrow Z$



Corrigeren gelijk tijdstipRegistratie

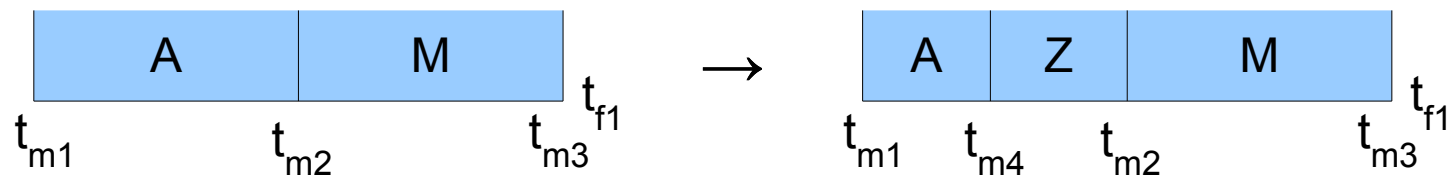
Erratum 5.4.2

$A(t_{m1}, t_{m2}) \rightarrow Z((t_{m4}, t_{m2}); t_{f1}),$

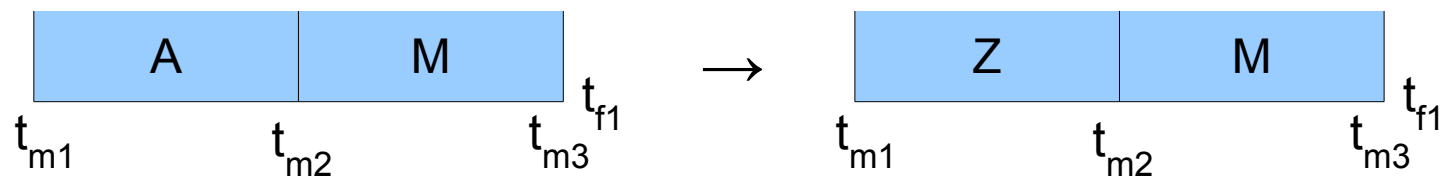
$A \neq Z$ en $t_{m2} \leq \infty$ en $t_{m1} \leq t_{m4} < t_{m2}$

A moet van waarde veranderen op t_{m1} en t_{m2}

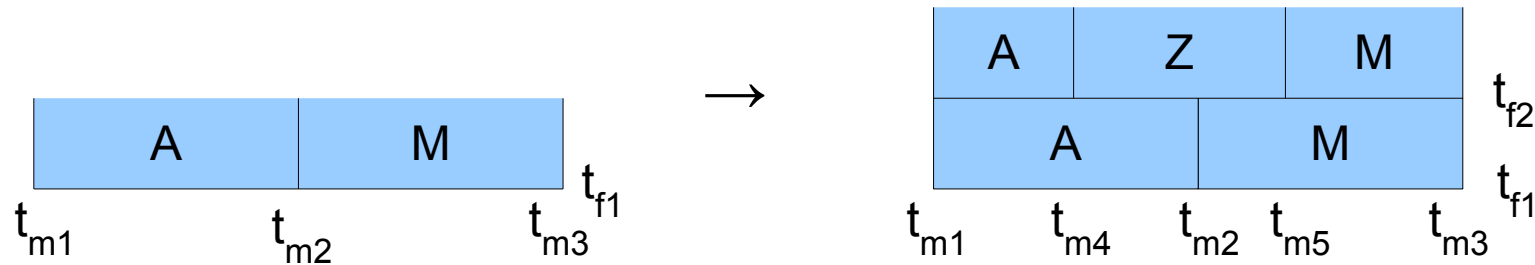
In database heeft A als tijdstipRegistratie t_{f1}



of

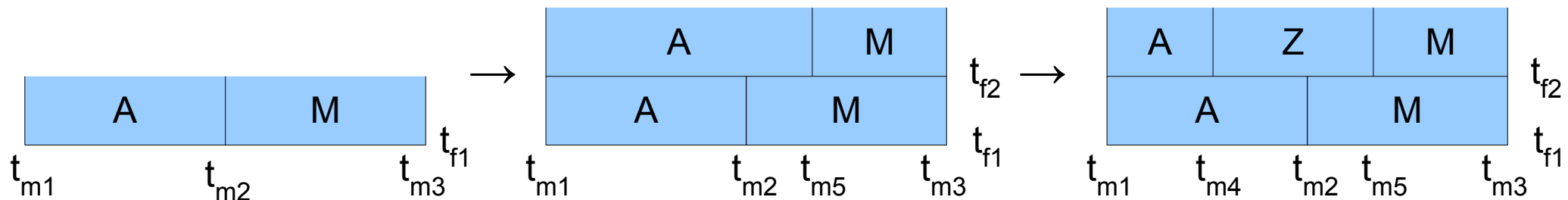


Alternatief tussenvoegen



$$A(t_{m1}, t_{m2}) \rightarrow Z((t_{m4}, t_{m5}); t_{f2})$$

Maar ook:



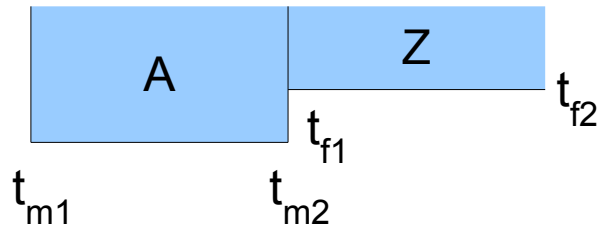
$$A(t_{m1}, t_{m2}) \rightarrow A((t_{m1}, t_{m5}); t_{f2}) + A(t_{m1}, t_{m5}) \rightarrow Z((t_{m4}, t_{m5}); t_{f2})$$

Is tussenvoegen nodig?

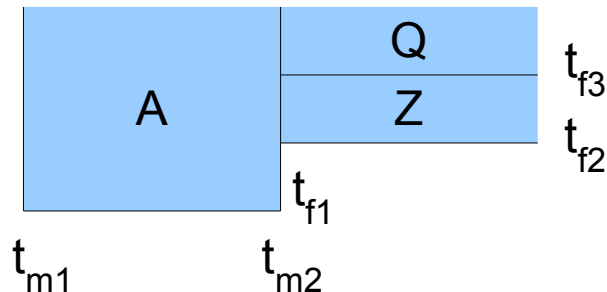
- Minder kennisgevingen (performance)
- Extra operatie ??? (implementatiekosten)
- Gemak opbouw synchronisatie bericht

In erratum 5.4.2 is tussenvoegen historie
geschrapt

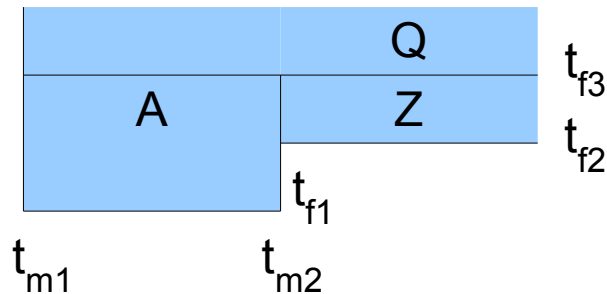
Opbouw synchronisatie (1)



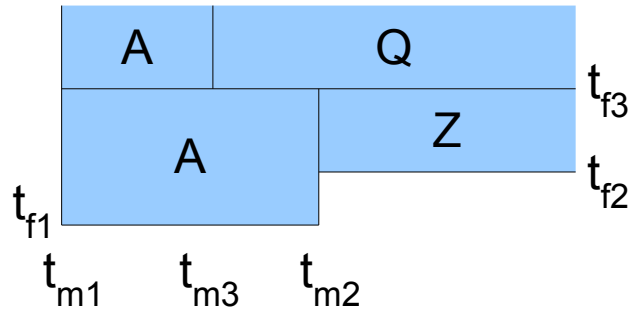
- Eén blok voor tijdstipRegistratie met eindGeldigheid ∞ zonder formele historie actueel
→ Wijzigkennisgeving actueel



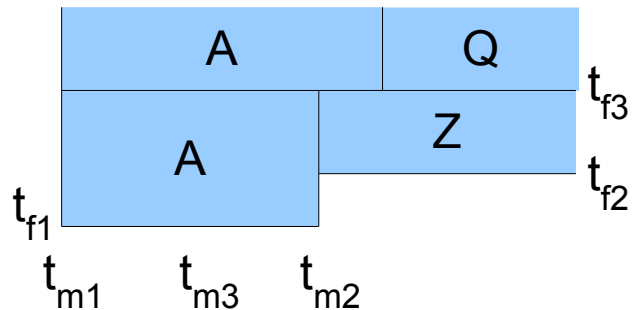
- Eén blok voor tijdstipRegistratie met eindGeldigheid ∞ met formele historie actueel
→ Correctiekennisgeving actueel met al dan niet wijziging beginGeldigheid



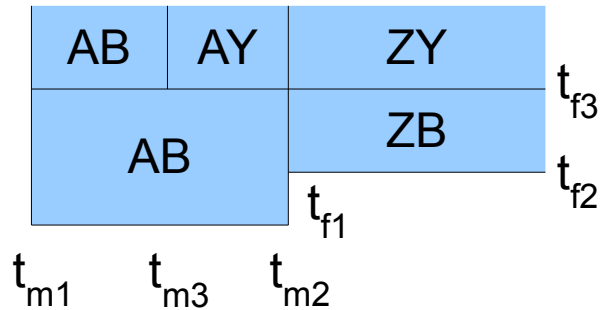
Opbouw synchronisatie (2)



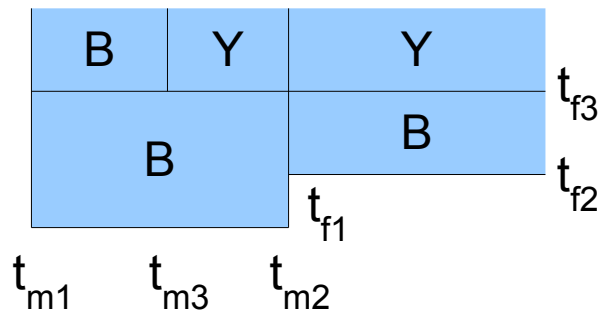
- Twee blokken voor tijdstipRegistratie
- Laatste blok eindGeldigheid ∞
- Eerste blok gelijk aan oudste formele historie
→ Correctiekennisgeving actueel met wijziging beginGeldigheid



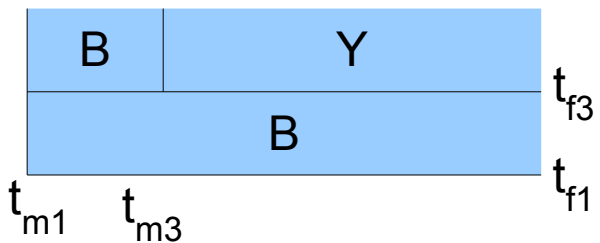
Opbouw synchronisatie (3)



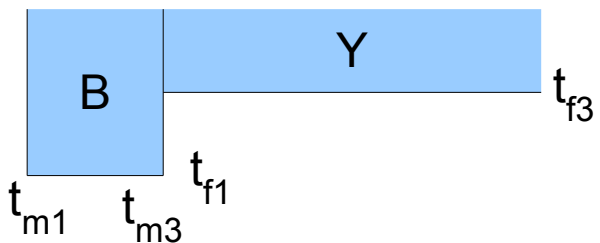
Verwijder alle waarden die in de formele historie niet verschillen van waarden in aangrenzende blokken



Verwijder irrelevante materiële en formele tijdstippen

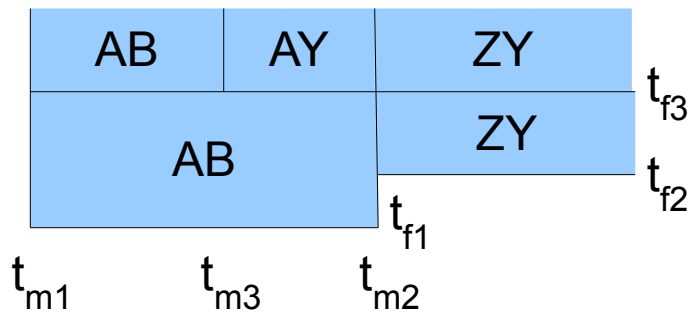


Schoon op tot linked list representatie



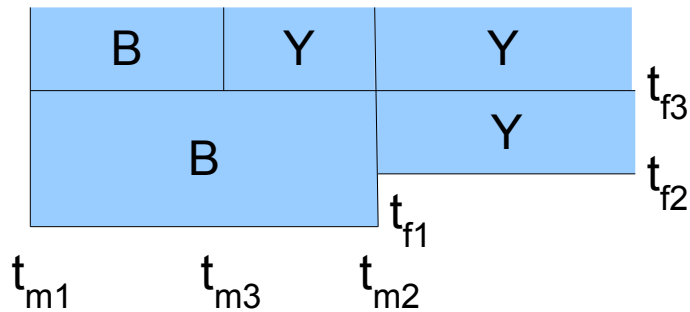
Maak actuele wijzigkennisgeving

Opbouw synchronisatie (4)

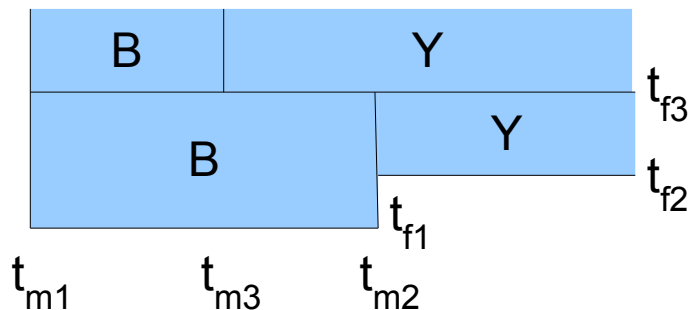


Verwijder alle waarden waarvoor geldt:

- (1) in de formele historie verschillen ze niet van waarden in aangrenzende blokken
- (2) in de materiële historie hebben ze op het voorgaande tijdstip Registratie (t_{f2}) een andere waarde dan in de materiële buurblokken

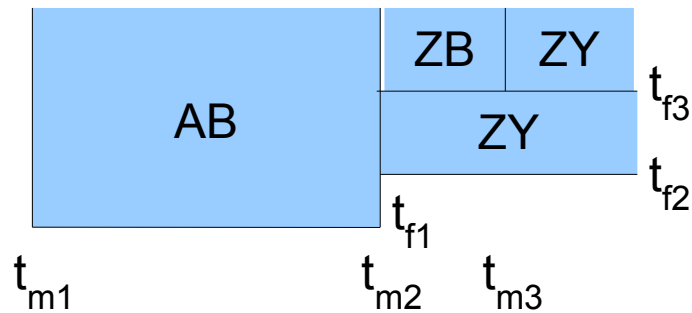


Verwijder irrelevante materiële en formele tijdstippen



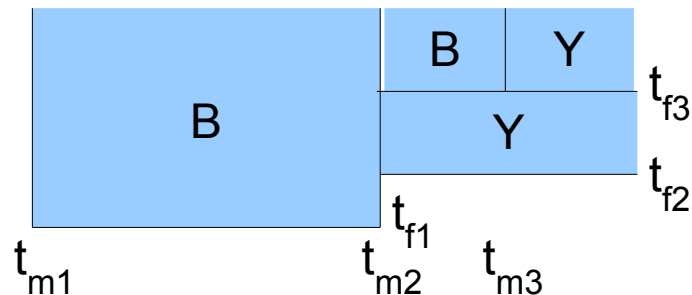
Maak actuele correctiekennisgeving met verschuiving beginGeldigheid

Opbouw synchronisatie (5)

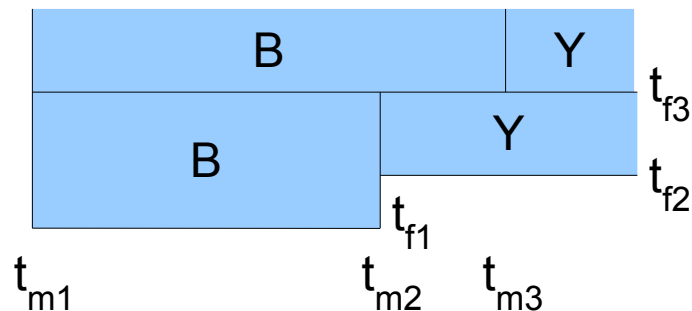


Verwijder alle waarden waarvoor geldt:

- (1) in de formele historiever schillen ze niet van waarden in aangrenzende blokken of wijzigen ze vanaf het voorgaande tijdstipRegistratie (t_{f2}) niet
- (2) in de materiële historie op het voorgaande tijdstipRegistratie (t_{f2}) hebben ze een andere waarde dan in de materiële buurblokken

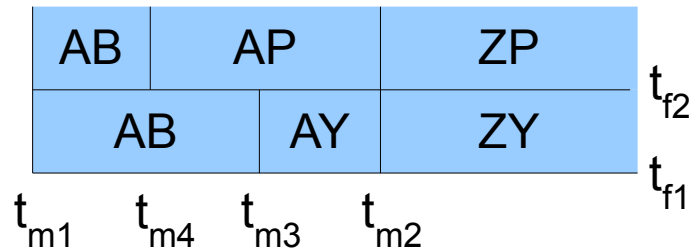


Zorg dat waarden voor hun grootste tijdstipRegistratie zo lang mogelijk zijn



Maak actuele correctiekennisgeving met verschuiving beginGeldigheid

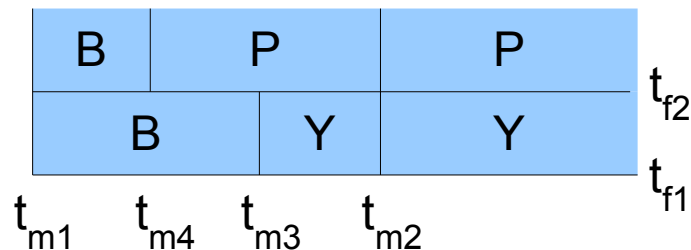
Opbouw synchronisatie (6)



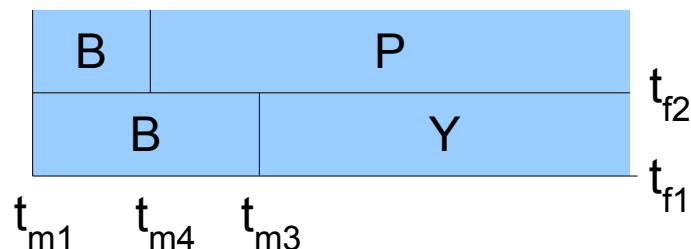
Verwijder alle waarden waarvoor geldt

(1) in de formele historie verschillen ze niet van waarden in aangrenzende blokken of wijzigen ze vanaf het voorgaande tijdstipRegistratie (t_{f2}) niet

(2) in de materiële historie hebben ze aan weerszijden van de grens op het huidige tijdstipRegistratie (t_{f2}) en het voorgaande tijdstipRegistratie (t_{f1}) allebei dezelfde waarde of allebei een andere waarde dan in de materiële buurblokken

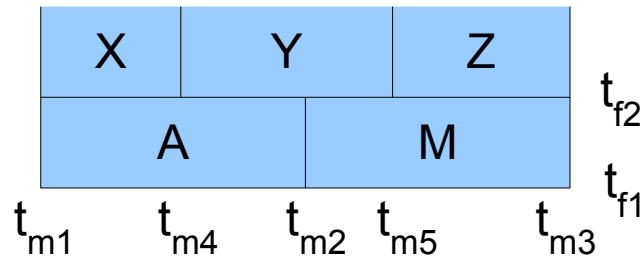


Verwijder irrelevante materiële en formele tijdstippen



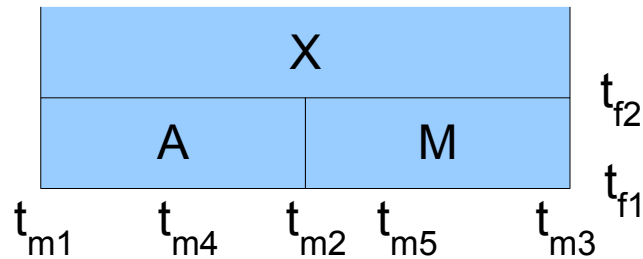
Maak actuele correctiekennisgeving met verschuiving beginGeldigheid en wijziging $Y \rightarrow P$

Historische correcties



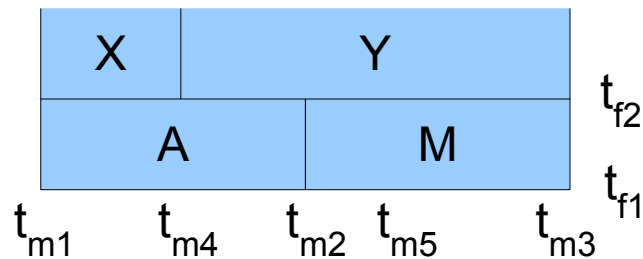
Geef de in te vullen ruimte in eerste instantie over de volle lengte de waarde X:

$$A(t_{m1}, t_{m2}) \rightarrow X((t_{m1}, t_{m3}); t_{f2})$$



Geef vanaf t_{m4} de waarde Y:

$$X(t_{m1}, t_{m3}) \rightarrow Y((t_{m4}, t_{m3}); t_{f2})$$



Geef vanaf t_{m5} de waarde Z:

$$Y(t_{m4}, t_{m3}) \rightarrow Z((t_{m5}, t_{m3}); t_{f2})$$

Het gaat hetzelfde als $Z = M$ maar kan efficiënter met

$$A(t_{m1}, t_{m2}) \rightarrow X((t_{m1}, t_{m5}); t_{f2}) + X(t_{m1}, t_{m5}) \rightarrow Y((t_{m4}, t_{m5}); t_{f2})$$

Als $Z = M$ en $X = A$ kan het ook met tussenvoegen (één kennisgeving)

Opbouw synchronisatie (7)

Gebruik mechanisme historische correcties in alle gevallen dat het versimpelen niet leidt tot een enkelvoudige kennisgeving

Problemen synchronisatie

- Synchronisatieberichten zijn erg groot
 - Herhalen stuurgegevens
 - Herhalen kerngegevens
 - Alle relaties → Relaties niet los corrigeerbaar
- Bij correctie beginRelatie/eindRelatie moet ook tijdvakGeldigheid gecorrigeerd worden
- Identificatie relaties lastig als ook kerngegevens gecorrigeerd zijn
 - Belanghebbenden met gecorrigeerd tijdvakRelatie of aanduidingEigenaarGebruiker

sleutelVerzendend voor identificatie

- Neem in synchronisatiebericht in elk object (fundamenteel en relatie) sleutelVerzendend op
 - Identificeer objecten obv synchronisatie actueel
 - beschikbaarheid (?): genereer zonodig id uniek binnen synchronisatiebericht
- Kleinere berichten
- Ook relaties altijd identificeerbaar

Relaties: apart synchronisatiebericht

- Relaties zijn zelfstandige objecten
- Gebruik StUF-structuur:
 - Parent (I) → Relatie (T, W, E, R) → gerelateerde (I)
- Er mogen meerdere relaties van hetzelfde type gecorrigeerd worden (correctie tijdvakRelatie)
- Per kennisgeving slechts één tijdstipRegistratie
- Sh03/Sh04 fundamenteel: blijft vragen om synchronisatie fundamenteel incl. alle relaties

Praktijk LV WOZ

Regelmatig te corrigeren actuele of historische gegevens

- WOZKOZ en SWOKOZ: toegekendeOppervlakte en meegetaxeerdeOppervlakte
- WOZSUB: aanduidingEigenaarGebruiker
- WRDSUB: statusBeschikking, brondocument

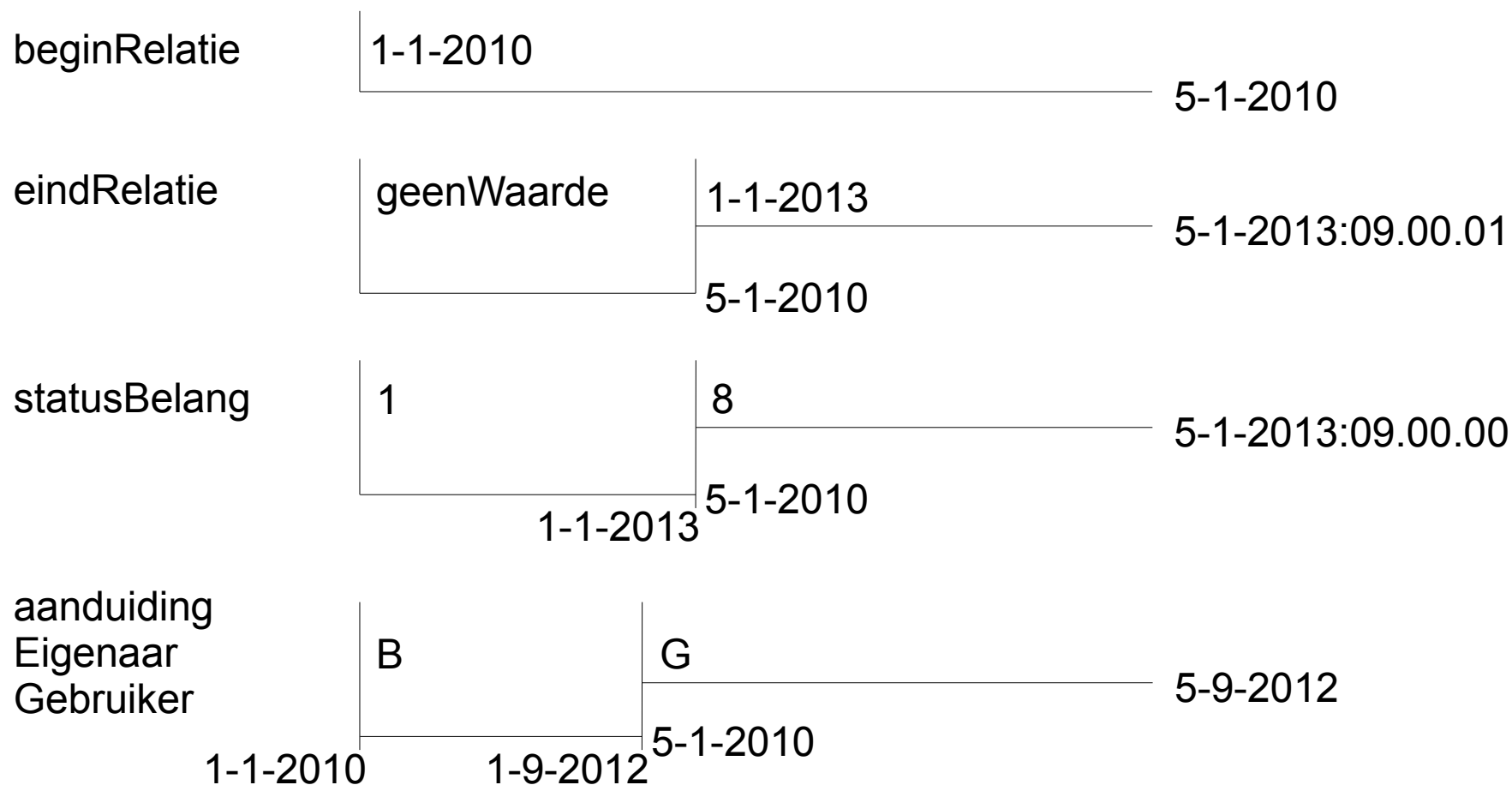
tijdvakRelatie: aansluitende relaties → synchronisatiebericht

Zijn correcties historie relevant (?): inOnderzoek

Komen correcties historie wel voor (?):

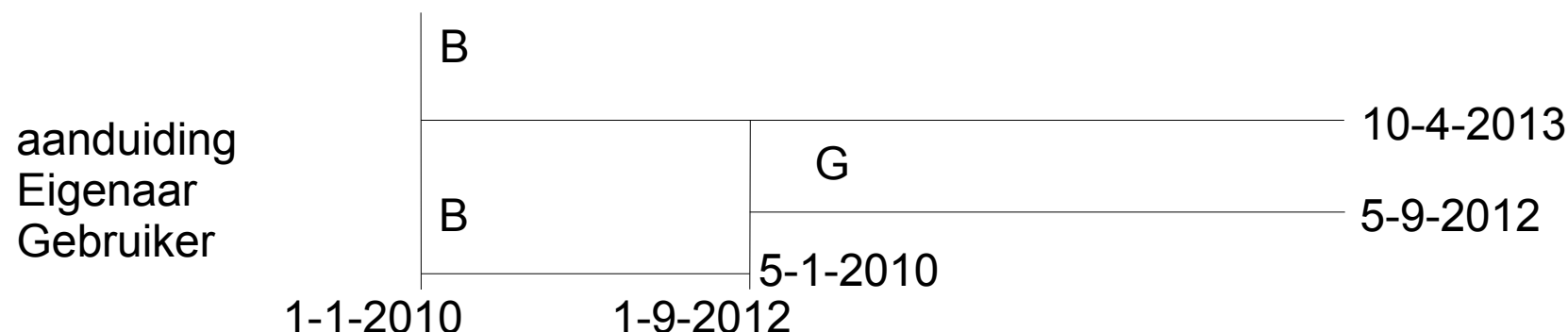
- WOZNUM: locatieOmschrijving
- WOZSUB: avr.aard
- WRDWSP: aandeelWaardeWaterschap

Verschuiven eindRelatie (1)

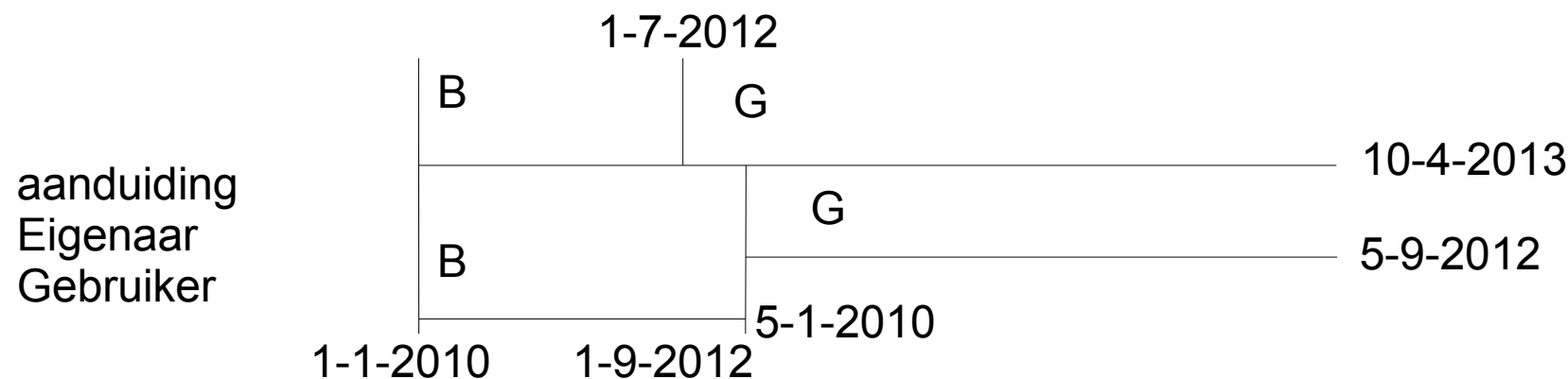


Verschuiven eindRelatie (2)

1: Waarde moet na verschuiven altijd B zijn



2: beginGeldigheid G verschuift ook



Verschuiven eindRelatie (3)

Verschuiving naar datum na wijziging aanduidingEigenaarGebruiker

	beginRelatie	eindRelatie	beginGeldigheid	eindGeldigheid	statusBelang
Oud	1-1-2010	1-1-2013	1-1-2013	geenWaarde	8
Nieuw	1-1-2010	1-11-2012	1-11-2012	geenWaarde	8

Verschuiving naar datum voor wijziging aanduidingEigenaarGebruiker

	aanduidingEigenaarGebruiker	beginGeldigheid	eindGeldigheid
Oud	G	1-9-2012	geenWaarde
Nieuw	B	1-1-2010	geenWaarde

+

	beginRelatie	eindRelatie	beginGeldigheid	eindGeldigheid	statusBelang
Oud	1-1-2010	1-1-2013	1-1-2013	geenWaarde	8
Nieuw	1-1-2010	1-8-2012	1-8-2012	geenWaarde	8

met gelijke tijdstipRegistratie

Verschuiven beginRelatie

- Neem naast wijzigende beginRelatie en beginGeldigheid in oud/nieuw alle elementen met dezelfde waarde op waarvan beginGeldigheid ook wijzigt
- Van wijzigende beginRelatie moet tijdvakGeldigheid kloppen
- tijdvakGeldigheid schuift ook voor elementen die wijzigen op beginGeldigheid oud
- Doe niets met elementen die niet wijzigen op beginGeldigheid (bijv. eindRelatie)

Tussenvoegen relatie

Eén synchronisatiebericht relatie met voor alle wijzigingen dezelfde tijdstipRegistratie

- Maak eerst ruimte door begin/eindRelatie te verschuiven van de relatie waarvoor/na de tussen te voegen relatie komt
- Voeg vervolgens de relatie tussen door hem toe te voegen

→ **Dit wijkt af van huidige specificatie**

Verkleinen berichten

- Verwijder overbodige stuurgegevens door gebruik van een element met functie="update"
 - Voorbeeldbericht wordt circa 25% kleiner
- Introduceer historische correctieberichten
 - Mogelijk door sleutelVerzendend tbv identificatie
 - Zelfde structuur als synchronisatieberichten (beginnen ook met synchronisatie actueel)
 - Bevatten precies de kennisgevingen tbv correctie

Andere visie op beheermodel

- StUF-standaard mag uitgebreid worden zonder namespace wijziging, mits bestaande sectormodellen niet worden geraakt
- Beheerder sectormodel kan in overleg met gebruikers beslissen om nieuwe functionaliteit op te nemen zonder namespace wijziging, mits bestaande functionaliteit niet wijzigt, maar slechts wordt uitgebreid
- Uitbreiding basis complexTypes niet toegestaan zonder namespace wijziging